DOCUMENT RESUME

ED 267 122 TM 860 215

AUTHOR Owston, Ronald D.; Dudley-Marling, Curt
TITLE A Criterion-Based Approach to Software Evaluation.
PUB DATE 17 Apr 86
NOTE 18p.; Paper presented at the Annual Meeting of the American Educational Research Association (70th, San Francisco, CA, April 16-20, 1986).
PUB TYPE Speeches/Conference Papers (150) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Software; Correlation; Elementary Secondary Education; Evaluation Criteria; *Evaluation Methods; Guidelines; *Models; *Reliability; Summative Evaluation; *Validity
IDENTIFIERS *Screening Procedures

ABSTRACT

The overall poor quality of educational software on the market suggests that educators must continue efforts to evaluate available packages and to disseminate their findings. In this paper, weaknesses in published evaluation procedures are identified, and an alternative model, the York Educational Software Evaluation Scale (YESES), is described. The rationale for this criterion-based model is drawn from the fields of the assessment of student writing, criterion-referenced testing, and the assessment of second language oral proficiency. Four charactertistics important for evaluation were identified from an analysis of published evaluation guidelines: (1) pedagogical content; (2) instructional presentation; (3) documentation; and (4) technical adequacy. Data are presented on the mean ratings of software evaluated with the model, scale intercorrelations, and indicators of its validity and reliability. Feedback indicates that YESES is best used as an initial screening device to narrow the choice of software to a manageable few that can be examined in detail, and as a summative evaluation instrument. (Author/PN)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED267122

A Criterion-Based Approach to Software Evaluation

Ronald D. Owston and
Curt Dudley-Marling
York University

A Paper Presented at the Annual Meeting of the
American Educational Research Association
April 17, 1986

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

R. D. Owston

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

TM 860 215

Abstract

The overall poor quality of educational software on the market suggests that educators must continue efforts to evaluate available packages and to disseminate their findings. In this paper, weaknesses in published evaluation procedures are identified, and an alternative, criterion-based model is described. The rationale for the model is drawn from the fields of the assessment of student writing, criterion-referenced testing, and the assessment of second language oral proficiency. Data are presented on the mean ratings of software evaluated with the model, scale intercorrelations, and indicators of its validity and reliability. (Keywords: courseware, evaluation models, software evaluation.)

A Criterion-Based Approach to Software Evaluation

Over the past few years there has been large increase in the number of software titles available that have either been developed specifically for the education market, or have been developed for other markets--such as home or business--but have some applicability to the education market. Educational Products Information Exchange (EPIE), for example, reported some 4500 titles in its 1984 software directory (EPIE, 1984); this number increased to 6600 in the 1985 directory (EPIE, 1985). While EPIE does not claim to list the complete universe of microcomputer educational software in its directories, nevertheless they are widely acknowledged as being one of the most complete sources of information on available software. Today there is an estimated 8000 plus educational titles available.

Unfortunately, the quality of educational software has not increased concomitantly. EPIE reported that only five percent of available educational software could be rated as "exemplary" (staff, 1984). While many knowledgeable educators found this figure unacceptably low, EPIE later confirmed its accuracy when they applied the new California State Department of Education "Guidelines for Educational Software for California Schools" to a representative sample of software (staff, 1984/85). Further support of this thesis comes from Alberta Education (1985) who found that they were unable to recommend about nine out of ten software products previewed for use in provincial schools. Bialo and Erikson (1985) concluded--after an analysis of all software evaluations carried out by EPIE--that most educational software currently being developed is poorly designed and does not take advantage of the potential or capabilities of the microcomputer. More recently, EPIE reported that an even more disturbing trend in software quality is appearing. An analysis of both EPIE and non-EPIE software evaluations for the period 1980-84 suggested that the overall level of quality had "stalled out" at the lower end of EPIE's "recommended with reservations" rating range during the last two years (staff, 1985).

Thus educators are in the unenviable position of having to select software from from an ever-increasing pool that appears to be maintaining a constant, low level of quality. Such a generalization may be unfair to an unknown number of individual products, however. There is little doubt that specific products exist that far surpass the quality of the best products developed only a few years ago. The task we are faced with quite simply is to identify these products. This then implies that earlier efforts to establish and

maintain software clearinghouses, and to encourage teachers to become involved in software evaluation, must be sustained. At the same time we must take a critical look at software evaluation procedures that have been used to date to see if they are meeting the needs of educators, and if they are not we must develop alternative procedures.

Current Software Evaluation Approaches

A survey of the literature will easily turn up some 40 to 50 different approaches that have been suggested for software evaluation. Baker (1983) has proposed a convenient model to view them. He suggests that they can be organized along a continuum according to the formality of the approach. Four main points on this continuum can be identified:

1. Organized networks with large numbers of evaluators using given sets of guidelines such as MicroSIFT and EPIE. Results of these evaluations are widely disseminated through directories, professional journals, and on-line databases.

2. Subscription publications such as Courseware Report Card and Software Reports that do not have formal networks of evaluators, but instead rely both on in-house and out-of-house evaluators.

3. Organizations such as the Minnesota Educational Computer Corporation (MECC), SOFTSWAP, and CONDUIT. The primary function of these organizations is software development, yet evaluation is an important component of the development process.

4. A category of discrete evaluation forms that individuals or groups may freely use. Typical of these forms are the one developed by the National Council of Teachers of Mathematics (Heck et al., 1984), those that have appeared in many computing periodicals, and those developed by school boards for local purposes.

Several problems can be identified with these approaches, however. First, current approaches tend to be normative in nature. That is, evaluators are asked to rate software according to their strength of agreement to statements about the software, or they are asked to give written opinions on various aspects of the software. For example,

Presentation of content is clear and logical.
SA A D SD NA
(International Council for Computers in Education, 1984, p. 18)

Or,

Is the program easy to run? YES NO Describe_____
(EPIE, 1982)

Since none of the current approaches are based on explicit criteria, evaluators tend to judge software relative to other software they have seen. If the state-of-the-art of software development were more advanced, this would not be a problem, because meaningful normative comparisons could be made. This is a distinct limitation though when the norm is considered to be inadequate by most educators. Evaluators will be able to respond to questions such as those above by saying, for example, "yes, the software is easy to run". But what is meant by "easy to run"? Easier than software XYZ? The difficulty is that software XYZ may not easy to run and, even though a new piece of software may be judged superior, the new software may still be inadequate.

Another problem is that current evaluation procedures tend to be subjective. This will not pose a problem if the evaluators are well-known authorities whose opinions are highly valued (Eisner, 1979). A problem does occur, though, if the evaluators are unknown to the evaluation reader, which is most often the case in software review columns in periodicals or other widely disseminated evaluations. We do not know what philosophy, beliefs, or biases the unknown evaluator is bringing to bear on the evaluation. Publications such as Software Reviews help reduce this kind of problem because the reader can compare the abstracts of several reviews to look for trends or an emergent consensus. Nevertheless, the basic subjective element remains.

A problem stemming from the lack of standards and subjectivity of current software evaluation approaches is their inherent lack of reliability. Current approaches make little or no attempt to assure the reader that their evaluations have some measure of reliability either within raters or between raters. Evaluation consumers will typically be faced with the task of having to choose one software package out of several packages that purportedly accomplish the same objectives. In the absence of any measure of consistency or reliability of the evaluation procedure being used, meaningful evaluative comparisons are very difficult to make. This

limitation is even more serious when different evaluators evaluate different packages.

The difficulty of obtaining an overall impression of a piece of software is another problem with present evaluation approaches. These approaches typically require the evaluator to answer as many as 40 or 50 questions about the software, yet provide little, if any, guidance in interpreting the discrete answers into a meaningful whole. Such guidance is necessary since ultimately the evaluation consumer is going to want to make an overall judgement about a particular package under consideration. Those evaluation approaches that do offer some guidance suggest that the evaluator (or evaluation consumer) take a global rating by summing the answers to individual questions in the evaluation instrument. For example, Bitter and Camuse (1984) suggest that evaluators assign their own weightings to each individual question and then calculate a weighted total score. Test (1985) suggests that the total number of YES and NO responses to his evaluation instrument be added to produce "the number of desirable characteristics in the program". Clearly, any such scheme is far too simplistic to take into account the complex interactions of the many important variables that combine to produce quality software. Without a meaningful overall impression the reader is not able to readily compare similar kinds of software when making instructional or purchasing decisions.

A Criterion-Based Evaluation Approach

The York Educational Software Evaluation Scales (YESES) were designed to overcome the weaknesses of current approaches (Owston, 1985a). Two requirements were of prime consideration for YESES when it was being developed. First, the scale had to reasonably concise because evaluations using the scale were to placed into the York Faculty of Education On-Line Service, a nationally available database of information on educational software. It was felt that the entire evaluation should occupy no more than about one screenful of text to avoid reader fatigue. And second, the evaluations had to meet the information requirements of the intended audiences, for as Guba (1978) points out, the most important criterion for the validity of an evaluation is the extent to which audience understanding is increased. In this case, the intended audiences were (1) teachers who may want to select software for classroom use from a central library and need evaluative comments to help narrow their search to a

manageable number of titles to preview, (2) educators who have to make software purchasing decisions, and (3) software producers who would like summative evaluations of their products.

The rationale for the design of YESES was drawn from three sources. The first is the field of the analytical assessment of student writing (Diederich, 1974). In this field, the notion exists that there are several identifiable underlying traits of writing that are considered to be important in any kind of writing in any context, upon which the writing can be judged. Furthermore, experts seldom have difficulty in agreeing on what most of these traits are (e.g. organization, ideas, mechanics, wording). In brief, a scale is developed to assess each of these traits, and the assessment of a piece of writing is reported in terms of a score for each of them. A second field from which the rationale for YESES was drawn was criterion-referenced testing (Popham, 1978). The belief in this field is that the most meaningful interpretations of test results come from comparing mastery relative to specified domains of knowledge, rather than from normative comparisons. By doing so one can find out what the learner actually knows, instead of simply finding out that the learner knows more (or less) than other learners. The third area the YESES rationale is drawn from is the field of the assessment of oral proficiency in a second language. In particular, the rationale comes from the developments pioneered by The Educational Testing Service with the U.S. Foreign Service Institute language examinations, and subsequently adapted by educational jurisdictions, one of which was the New Brunswick Department of Education (1974) in Canada. The assessment procedure requires interviewers to be trained and "calibrated" to a holistic proficiency scale. Then through a structured conversation, the interviewer is able to locate the overall proficiency of the interviewee at an appropriate point on the scale. The scale is interpreted by referring to sets of descriptors that describe in detail the language skills typical of an individual at that point.

In developing YESES four characteristics that evaluators and evaluation consumers believed important for the evaluation of drill and practice, tutorial, problem solving, and simulation software were identified from an analysis of published evaluation guidelines. They were the pedagogical content, instructional presentation, documentation, and technical adequacy. Empirical support for the salience of these characteristics was obtained by Marshall and Cannings (1984) who, using the Delphi technique and seven published evaluation checklists, asked panels of educators to generate

and later confirm the most important attributes for the evaluation of software. In addition to the four characteristics above, a fifth characteristic, modelling, was identified to evaluate simulation software. Although no such characteristic has been used for software evaluation, it was felt that since simulation software has a valuable role to play in the classroom, a unique evaluation criterion would encourage the use and development of this kind of software. The resulting five evaluation characteristics were then defined in detail in terms of what they were and what they were not. For each characteristic, a four point criterion-based scale was developed with the points representing "exemplary" software, "desirable" software, "minimally acceptable" software, and "deficient" software. Each point on the scale was defined by a set of descriptors that give typical characteristics of software that would be rated at that level. Thus, with YESES, the process of evaluation is one of determining which set of descriptors best characterizes the software on each of the four or five scales.

The resultant evaluation scales were next circulated to colleagues and teachers experienced in the use of microcomputers in education for criticisms and suggestions. A revised form of YESES was then subjected to formal use and subsequent revisions. Although the final form of YESES is too lengthy to reproduce in this paper, the content scale definition and categories are given in the appendix to illustrate the nature of the the scales.

Panel Evaluation

YESES is used as the evaluation instrument in a model known as panel evaluation. As the name implies, panels of evaluators are convened to evaluate software. Each panel consists of two or three members drawn from a pool of classroom teachers, a subject area consultants, and university faculty members. Important to note is that computer consultants or others with computer expertise are not necessarily sought to become panel members. While computer-related skills are valuable, the main criterion for panel membership is sound expertise in the teaching/learning process.

Panel members are first trained in the use of YESES before conducting any evaluations. This training involves having the panels blindly rate a "range finder" piece of software that has previously been rated by the scale developers, and then share their ratings with the trainers to resolve any

discrepancies or misunderstandings about the scale. Two modes of operation for the panels have been used. One has each panel member first rate a given piece of software independently. Then, as a group, the panel arrives at a consensus about what the final ratings of the software should be. The other mode has the panel as a team, jointly examine the software and develop a consensus along the way. (Because of evaluator preferences, the latter mode has most often been used.) After the final ratings for the software on each scale have been determined, the panel is asked to write a short (less than 200 words) narrative describing any unusual features of the software, suggesting unique ways which it may be used, explaining any particularly extreme ratings, or noting special conditions under which the evaluation took place.

Panel Evaluation Results

Over 100 educational software titles have been evaluated using YESES and the panel evaluation model, representing a wide variety of software types and publishers. Summary statistics, scale intercorrelations, and an indication of the validity of YESES were reported by Owston (1985b) in a study of the first 57 evaluations conducted.

The mean rating on the content scale for this sample was 2.19 (standard deviation .93). Fully 93 percent of the software evaluated was rated "deficient", "minimally acceptable", or "desirable". As the mean and standard deviation suggest, these ratings were spread quite uniformly over the three scale levels. The remaining 7 percent of the software evaluated was rated "exemplary". The mean rating for the instruction scale was 2.28 (standard deviation .88), with slightly more of the software evaluated being rated as "desirable" or less (95 percent), and slightly less being rated as "exemplary" (5 percent). The mean rating for the technical adequacy scale was 2.54, higher than both the content and instruction scales. The scale standard deviation was .83. Fewer software packages were rated "desirable" or less (91 percent), and more were rated "exemplary" (9 percent) on the technical scale than on the previous two scales. Software on the documentation scale was rated overall lower than technical, but only sightly higher than instruction and content (mean 2.30, standard deviation .84). The same proportion of software was rated "desirable" or lower and "exemplary" on the documentation scale as on the technical scale. Of the 57 software packages reported on by Owston,

only six were simulation, thus any conclusions about the modelling scale are very tentative. The mean rating of this software was 3.00 and the standard deviation was 1.10. Thirty-three percent of the software was rated "exemplary", 50 percent as "desirable", and 17 percent as "deficient". None of the software was rated "minimally acceptable".

As mentioned earlier, the intercorrelations of the five scales of YESES were computed. Modelling correlated the highest with all other scales ranging from .83 with documentation to .71 with technical adequacy. Other correlations ranged from a high of .62 between technical and content to a low of .28 between technical and documentation. These correlations suggest that, except for modelling, all of the remaining four scales of YESES are reasonably independent. Although the sample size for modelling is small, the need for a separate scale for modelling is questionable and should be the subject of further research.

Qualitative comparisons were made between EPIE (n.d.) and YESES evaluations to obtain an indication of the validity of the panel evaluation approach. EPIE was selected as an appropriate criterion for a validity study because their evaluations are widely disseminated, and because the EPIE model involves trained evaluators to assure consistency in evaluation standards. Overall the qualitative analysis, which included both written comments and scale ratings, suggested a good level of agreement between YESES and EPIE. Seventeen titles had been evaluated by both models. The two evaluation approaches seemed to be in general agreement in ten of the seventeen cases, and to be in disagreement in another five. The evaluations of two other pieces of software showed partial agreement.

When YESES and EPIE agreed on the overall quality of a product, whether this be high or low, they frequently criticized different features of the package. One review might question the manner in which a pedagogical approach was implemented, while the other might be critical of the educational value of the program's content. In cases where there was broad disagreement about the overall quality of the package, the more negative review was as likely to be critical of the educational value of the activity as it was to be critical of the way which the activity had been implemented. Usually it would not be critical of both. Furthermore when YESES and EPIE disagreed on the overall quality of a product, the greatest discrepancies occurred with language arts software in the areas of content and instruction.

No formal study has been done on the reliability of the panel evaluation model. From time to time, however, the same title has been given to different panels for evaluation. This has been done both on the same day, and also with several months between. Every time different panels have evaluated the same piece of software, panels have been either in total agreement on the evaluation ratings, or they have disagreed by no more than one point on one, two, or three scales. In no cases have panels disagreed on all four or five scales.

Discussion and Conclusions

YESES, together with the panel evaluation model, was designed to improve on current instruments and practices in software evaluation. The weaknesses identified with these practices include their normative nature, subjectiveness, lack of reliability, and difficulty in obtaining an overall impression.

To some extent YESES appears to have been successful in lessening these concerns. For its part, the normative element in YESES still exists. Clearly, the scale level descriptors could not be established without reference to the current state of educational software. These descriptors will undoubtedly have to be revised when we start moving into the next generation of software. Furthermore, it is not realistic to expect evaluation panels to rate software without being influenced by ratings they have given to previously examined software. The influence can be minimized, however, by cautioning panels to always keep referring to the criteria of the scales.

The subjective element, while certainly not eliminated, appears to have been lessened with YESES and the panel evaluation approach. This is because the model has explicit evaluation criteria and because panel members must arrive at a consensus on the final ratings of the software. A measure of subjectivity may still occur though in the interpretation of the evaluation criteria, as no two evaluators will interpret them identically, or if one panel member tries to impose his or her beliefs about a specific piece of software on other panel members without regard to the criteria.

Reliability appears to be reasonably high with YESES. As mentioned earlier, panel ratings are very similar when different panels rate the same piece of software with or with-

out an intervening interval. Since none of the other existing evaluation models appear to have addressed this concern, comparisons are not possible, but there is little doubt that YESES is an improvement because of the panel evaluation training procedure and the use of explicit criteria. Nevertheless, formal studies need to be conducted to gain further insight into the reliability of YESES.

The remaining criticism of other approaches, concerning the difficulty of obtaining an overall impression of the evaluated software, seems reasonably well addressed by YESES. Perhaps unfortunately, the price paid for gaining an overall impression is losing evaluative information about specific details of the software. Therefore, YESES is best used as an initial screening device to narrow the choice of software down to a manageable few that can be examined in detail, and as a summative evaluation instrument.

Overall, feedback from educators experienced in using other evaluation approaches, as well as YESES, has been very positive. Many have said that they were able to learn more about a piece of software, in a relatively short period, with YESES, than with other approaches. Two reasons seem to account for this. First, being able to critically discuss software with colleagues and having to arrive at a consensus about the software provides a valuable learning experience for panel members. And second, when evaluators are forced to look at the software more as a whole, they avoid the trap provided by checklists of failing to see how the various elements of software interact and what the total impact of the software might be on the user. Thus from both a technical and a professional development point of view, YESES appears to be a viable software evaluation approach.

References

Alberta Education. (1985). Computer courseware evaluations: January, 1983 to May, 1985. Edmonton, Alberta: Author.

Baker, N. B. (Ed.). (1983). Evaluation of educational software: a guide to guides. Chelmsford, MA: The Northeast Regional Exchange.

Bialo, E. R. & Erikson, L. B. (1985). Microcomputer courseware: characteristics and design trends. AEDS Journal, 18,(4), 227-236.

Bitter, G. G. & Camuse R. A. (1984). Using a microcomputer in the classroom. Reston, VA: Prentice-Hall.

Diederich, P. B. (1974). Measuring growth in English. Urbana, IL: National Council of Teachers of English.

Eisner, E. (1979). The educational immagination. New York: Macmillian.

Educational Products Information Exchange. (1982). Microcomputer courseware evaluation form. Water Mills, NY: Author.

Educational Products Information Exchange. (1984). The educational software selector. Water Mills, NY: Author.

Educational Products Information Exchange. (1985). The educational software selector. Water Mills, NY: Author.

Educational Products Information Exchange. (n.d.). Micro-courseware PRO/FILES. Water Mill, NY: Author.

Guba, E. G. (1978). Toward a methodology of naturalistic inquiry in educational evaluation. Los Angeles: Center for the Study of Evaluation, UCLA.

Heck, W. P., Johnson, J., Kamsky, R. J., & Dennis, D. (1984). Guidelines for evaluating computerized instructional materials. Reston, VA: National Council of Teachers of Mathematics.

International Council for Computers in Education. (1984). Evaluator's Guide for Microcomputer-Based Instructional Packages. Eugene, OR: University of Oregon.

Marshall, K. W. & Cannings, T. R. (1984, April). Evaluating educational software: a comprehensive model. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

New Brunswick Department of Education. (1974). Manual for interviewers of French. Princeton, NJ: The Educational Testing Service.

Owston, R. D. (1985a). The York educational software evaluation scales. (York/IBM Cooperative Project Document 2). North York, Ontario: York University, Faculty of Education.

Owston, R. D. (1985b, December). Software evaluation using YESES. Paper presented at the annual conference of the Ontario Educational Research Council, Toronto, Ontario.

Popham, W. J. (1978). Criterion-referenced measurement. Englewood Cliffs, NJ: Prentice-Hall.

Staff. (1984, October). Research indicates stalled software evolution. MICROgram, pp. 3-6.

Staff. (1984/85, December/January). California sets software guidelines. MICROgram, p. 2.

Staff. (1985, April/May/June). What's in the educational software pool? MICROgram, pp. 1-4.

Test, D. W. (1985). Evaluating educational software for the microcomputer. Journal of Special Education Technology, 7(1), 37-46.

Appendix

CONTENT

Definition

Content refers to the knowledge and skills the software purports to teach--their organization, accuracy, and appropriateness. Content organization includes such aspects as the sequencing of the knowledge and skills within the lesson or lessons, the breadth or scope of the skills and knowledge, and the depth or intensity of instruction or practice given to a topic. Accuracy is concerned with truthfulness of the knowledge and skills that are presented. Appropriateness deals with the suitability of the content for the intended user which includes such factors as readability of the content, the match between the complexity of the content and the intended user's ability to master it, and the educational value of the content-- the time spent learning the content is justified because of its inherent value. The extent to which one, or all of these factors--organization, accuracy, and appropriateness--is weak is an indication of less than exemplary content.

LEVEL 4: Exemplary content

Level 4 content is superior in its organization, accuracy, and appropriateness. The content organization is such that the scope of the knowledge and skills is congruent with the user's ability to master them, the sequencing is logical and and follows good pedagogical practice (e.g. concrete presented before abstract), and the depth is sufficient to give the user adequate practice before proceeding to the next topic. The accuracy of level 4 content is extremely high. Furthermore, the content at this level is very readable, well-matched to the intended users's abilities, and has high educational value.

LEVEL 3: Desirable content

The organization, accuracy, and/or appropriateness of level 3 content is not quite as favourable as that of level 4 due to relatively minor weaknesses. The organization may be weak because the scope is not quite congruent with the user's ability to master it, the sequencing may be slightly illogical in several places or not quite in keeping with accepted pedagogical practice, or the depth may be either slightly more than necessary thus requiring the user to complete redundant exercises, or the depth may not be great enough so that the user does not receive sufficient practice before moving to the next topic. Problems with accuracy might consist of questionable (but not incorrect) facts or applications of concepts. Another possible class of difficulties with level 3 content is that there may be some vocabulary or sentence structures that may give some intended users difficulty, the knowledge or skills may be slightly too complex or too easy for the intended user, or some aspects of the content may be of slightly questionable educational value.

LEVEL 2: Minimally acceptable content

Level 2 content is clearly weak in either one, or a combination of, organization, accuracy, or appropriateness. The deficiency, however, is not serious enough to prevent the use of the software if no other better software is available and if the instructor is able to intervene to rectify the deficiency. Typical organizational problems found with level 2 software include the scope much greater than the user is able to deal with comfortably, the sequence poorly arranged or not consistent with good educational practice, or the depth considerably more or less than necessary. The kinds of accuracy problems encountered with level 2 content include incorrect minor facts or applications of concepts. The appropriateness problems found at this level include vocabulary and structure too difficult for most intended users, the knowledge and skills too difficult (or too easy), or the educational value of the content as a whole may be questionable.

LEVEL 1: Deficient content

Content at level 1 is sufficiently deficient so as to call into question the use of the software, regardless of the strengths of its other characteristics. Organizational problems may include weak, illogical sequencing, and scope and/or depth poorly matched with the user's ability. This level of content may also contain factual inaccuracies or incorrect applications of concepts. The content may not be very appropriate due to the reading level being considerably out of match with the user's ability, the knowledge and skills much too complex or simple, or the topics introduced by the software may be of very dubious educational value.